Intel®Pentium® M Processor on 90-nm Process with 2-MB L2 Cache

Specification Update

December 2004

`

Notice: The Intel® Pentium® M Processor may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are documented in this Specification Update.

Document Number: 302209-006

Contents

Revision History

Revision	Description	Date
-001	• Initial Release	May 2004
-002	• Updated Processor Identification Table	June 2004
-003	• Updated Processor Identification Table • Added Erratum X10	July 2004
-004	• Added Erratum X11, X12 and X13	October 2004
-005	• Added Erratum X14, X15 and X16	November 2004
-006	• Added Erratum X17 and X18	December 2004

§

Preface

This document is an update to the specifications contained in the documents listed in the following Affected Related Documents table. It is a compilation of device and document errata and specification clarifications and changes, and is intended for hardware system manufacturers and for software developers of applications, operating system, and tools.

Information types defined in the Nomenclature section of this document are consolidated into this update document and are no longer published in other documents. This document may also contain information that has not been previously published.

Affected/Related Documents

Document Title	Document Number
The Intel® Pentium® M Processor on 90nm process with 2-MB L2 Cache Datasheet	302189
IA-32 Intel Architecture Software Developer's Manual	
Volume 1: Basic Architecture	253665
Volume 2A: Instruction Set Reference	253666
Volume 2B: Instruction Set Reference	253667
Volume 3: System Programming Guide	253668

Nomenclature

S-Spec Number is a five-digit code used to identify products. Products are differentiated by their unique characteristics, e.g., core speed, L2 cache size, package type, etc. as described in the processor identification information table. Care should be taken to read all notes associated with each S-Spec number.

Errata are design defects or errors. Errata may cause the behavior of the Intel® Pentium® M processor on 90nm Process with 2-MB L2 Cache, to deviate from published specifications. Hardware and software, designed to be used with any given processor, must assume that all errata documented for that processor are present on all devices unless otherwise noted.

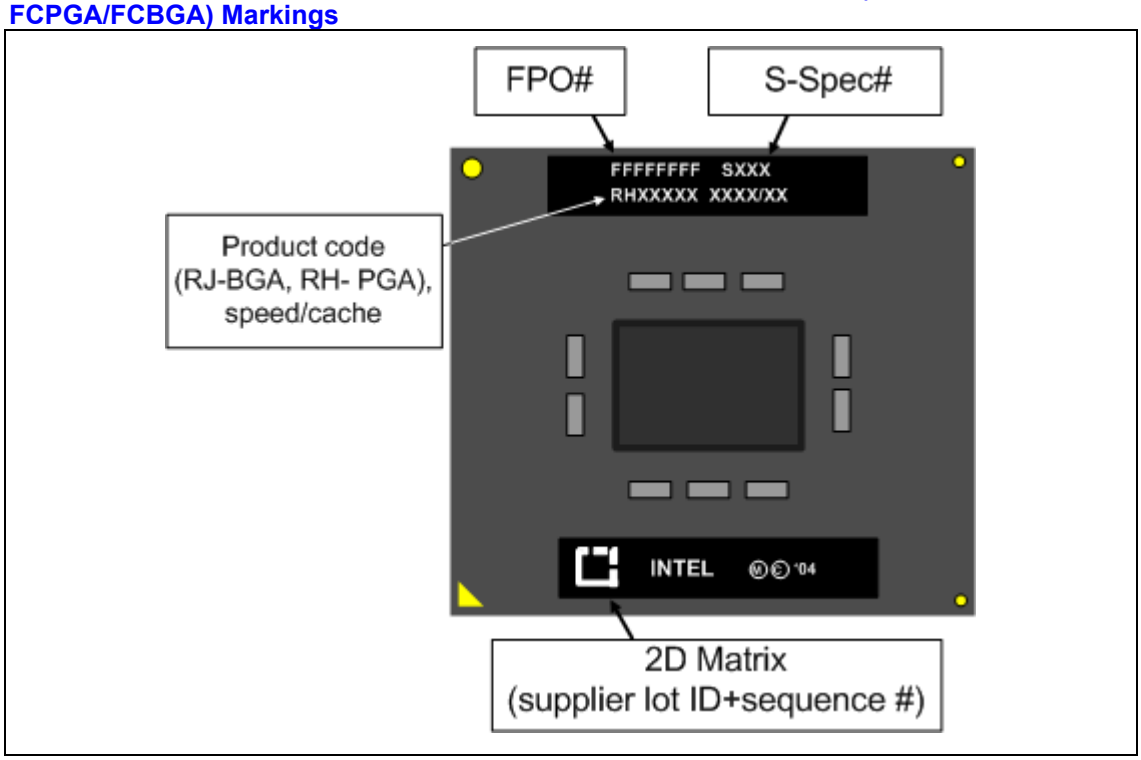
Documentation Changes include typos, errors, or omissions from the current published specifications. These changes will be incorporated in the next release of the specifications.

Specification Clarifications describe a specification in greater detail or further highlight a specification's impact to a complex design situation. These clarifications will be incorporated in the next release of the specifications.

Specification Changes are modifications to the current published specifications for the Intel Pentium M processor. These changes will be incorporated in the next release of the specifications.

General Information

Figure 1. Intel® Pentium® M Processor on 90nm Process with 2-MB L2 Cache (Micro-FCPGA/FCBGA) Markings



Identification Information

The Intel Pentium M processor on 90 nm process with 2-MB L2 Cache can be identified by the following values:

Family[1]	Model[2]	Brand ID[3]
0110	1101	00010110

-
  - The Family corresponds to bits [11:8] of the EDX register after Reset, bits [11:8] of the EAX register after the CPUID instruction is executed with a 1 in the EAX register.
  - The Model corresponds to bits [7:4] of the EDX register after Reset, bits [7:4] of the EAX register after the CPUID instruction is executed with a 1 in the EAX register.
  - The Brand ID corresponds to bits [7:0] of the EBX register after the CPUID instruction is executed with a1 in the EAX register.

Table 1. Identification Table

S-spec	Processor number	Product Stepping	CPU Signature	Core Speed		Package	QDF Notes
				Highest Frequency Mode (HFM)	Lowest Frequency Mode (LM)		
SL7V3	765	B-1	06D6h	2.1 GHz	600 MHz	Micro-FCPGA	6,2
SL7UZ	765	B-1	06D6h	2.1 GHz	600 MHz	Micro-FCBGA	6,2
SL7EM	755	B-1	06D6h	2.0 GHz	600 MHz	Micro-FCPGA	1,2
SL7EL	755	B-1	06D6h	2.0 GHz	600 MHz	Micro-FCBGA	1,2
SL7EN	745	B-1	06D6h	1.8 GHz	600 MHz	Micro-FCPGA	1,2
SL7EQ	745	B-1	06D6h	1.8 GHz	600 MHz	Micro-FCBGA	1,2
SL7EP	735	B-1	06D6h	1.7 GHz	600 MHz	Micro-FCPGA	1,2
SL7ER	735	B-1	06D6h	1.7 GHz	600 MHz	Micro-FCBGA	1,2
SL7EG	725	B-1	06D6h	1.6 GHz	600 MHz	Micro-FCPGA	1,2
SL7F2	725	B-1	06D6h	1.6 GHz	600 MHz	Micro-FCBGA	1,2
SL7GL	715	B-1	06D6h	1.5 GHz	600 MHz	Micro-FCPGA	1,2
SL7GK	715	B-1	06D6h	1.5 GHz	600 MHz	Micro-FCBGA	1,2
SL7VC	738	B-1	06D6h	1.4 GHz	600 MHz	Micro-FCBGA	3,2
SL7VD	733	B-1	06D6h	1.1 GHz	600 MHz	Micro-FCBGA	5,4
SL7V2	723	B-1	06D6h	1.0 GHz	600 MHz	Micro-FCBGA	5,4

NOTES

1. Multiple HFM VID's; VCC_CORE = 1.340-1.276 V for Highest Frequency Mode (HFM).
   - VCC_CORE = 0.988 V for Lowest Frequency Mode (LFM).
   - VCC_CORE = 1.116 V for Highest Frequency Mode (HFM).
   - VCC_CORE = 0.812 V for Lowest Frequency Mode (LFM).
   - VCC_CORE = 0.940 V for Highest Frequency Mode (HFM).
   - Multiple HFM VID's; VCC_CORE = 1.356-1.308 V for Highest Frequency Mode (HFM)

§

Summary Tables of Changes

The following table indicates the Errata, Documentation Changes, Specification Clarifications, or Specification Changes that apply to the processor. Intel intends to fix some of the errata in a future stepping of the component and to account for the other outstanding issues through documentation or specification changes as noted. This table uses the following notations:

Codes Used in Summary Table

Stepping

X:	Erratum, Specification Change or Clarification that applies to this stepping.
(No mark) or (Blank Box):	This erratum is fixed in listed stepping or specification change does not apply to listed stepping.

Status

Doc:	Document change or update that will be implemented.
PlanFix:	This erratum may be fixed in a future of the product.
Fixed:	This erratum has been previously fixed.
NoFix:	There are no plans to fix this erratum.

Row

Shaded:	This item is either new or modified from the previous version of the document.

Each specification update item is prefixed with a capital letter to distinguish the product. The key below details the letters that are used in Intel's microprocessor Specification Updates:

A = Intel® Pentium® II processor
B = Mobile Intel® Pentium® II processor
C = Intel®Celeron® processor
D = Intel® Pentium® II Xeon™ processor
E = Intel® Pentium® III processor
G = Intel® Pentium® III Xeon™ processorH = Mobile Intel® Celeron® processor at 466/433/400/366/333/300 and 266 MHz

K = Mobile Intel® Pentium® III processor

M = Mobile Intel® Celeron® processor

N = Intel® Pentium® 4 processor

O = Intel® Xeon™ processor MP

P = Intel® Xeon™ processor

Q = Mobile Intel® Pentium® 4 Processor supporting Hyper-Threading Technology on 90-nm process technology

R = Intel® Pentium® 4 processor on 90 nm process

T = Mobile Intel® Pentium® 4 Processor-M

V = Mobile Intel® Celeron® processor on .13 Micron Process in Micro-FCPGA Package

W =Intel® Celeron® M processor

X = Intel® Pentium® M processor on 90nm process with 2-MB of L2 Cache

Y = Intel® Pentium® M processor

Z = Mobile Intel® Pentium® 4 processor with 533 MHz Front Side Bus

The Specification Updates for the Pentium processor, Pentium Pro processor, and other Intel products do not use this convention.

NO.	B1	Plans	ERRATA
X1	x	No Fix	Code Segment is wrong on SMM Handler when SMBASE is not aligned
X2	x	No Fix	IFU/BSU Deadlock May Cause System Hang
X3	x	No Fix	Memory Aliasing with Inconsistent A and D Bits may Cause Processor Deadlock
X4	x	No Fix	RDMSR or WRMSR to Invalid MSR Address May Not Cause GP Fault
X5	x	No Fix	Unable To Disable Reads/Writes to Performance Monitoring Related MSRs
X6	x	No Fix	Move to Control Register Instruction May Generate a Breakpoint Report
X7	x	No Fix	Error in Instruction Fetch Unit (IFU) Can Result in an Erroneous Machine Check-Exception (#MC)
X8	x	No Fix	Code Fetch Matching Disabled Debug Register May Cause Debug Exception
X9	x	No Fix	Upper Four PAT Entries Not Usable With Mode B or Mode C Paging
X10	x	No Fix	SSE/SSE2 Streaming Store Resulting in a Self-Modifying Code (SMC) Event May Cause Unexpected Behavior
X11	x	No Fix	Code Segment limit violation may occur on 4 Gigabyte limit check
X12	x	No Fix	FST Instruction with Numeric and Null Segment Exceptions may cause General Protection Faults to be Missed and FP Linear Address (FLA) Mismatch
X13	X	No Fix	Code Segment (CS) is Incorrect on SMM Handler when SMBASE is not Aligned
X14	x	No Fix	Page with PAT (Page Attribute Table) Set to USWC (Uncacheable Speculative Write Combine) While Associated MTRR (Memory Type Range Register) is UC (Uncacheable) May Consolidate to UC
X15	x	No Fix	Under Certain Conditions LTR (Load Task Register) Instruction May Result in System Hang

NO.	B1	Plans	ERRATA
X16	x	No Fix	Loading from Memory Type USWC (Uncacheable Speculative Write Combine) May Get Its Data Internally Forwarded from a Previous Pending Store
X17	x	No Fix	FXSAVE after FNINIT Without an Intervening FP (Floating Point) Instruction May Save Uninitialized Values for FDP (x87 FPU Instruction Operand (Data) Pointer Offset) and FDS (x87 FPU Instruction Operand (Data) Pointer Selector)
X18	x	No Fix	FSTP (Floating Point Store) Instruction Under Certain Conditions May Result In Erroneously Setting a Valid Bit on an FP (Floating Point) Stack Register

NO.	B1	Plans	SPECIFICATION CHANGES
			No Changes

NO.	B1	Plans	SPECIFICATION CLARIFICATION
			None

NO.	B1	Plans	DOCUMENTATION CHANGES
			No Changes

§

Errata

X1. Code Segment Is Wrong on SMM Handler when SMBASE Is Not Aligned

Problem: With SMBASE being relocated to a non-aligned address, during SMM entry the CS can be improperly updated, which can lead to an incorrect SMM handler.

Implication: This is a rare condition that may result in a system hang. Intel has not observed this erratum with any commercially available software or system.

Workaround: Align SMBASE to 32 kB.

Status: For the steppings affected, see the *Summary of Table of Changes*

X2. IFU/BSU Deadlock May Cause System Hang

Problem: A lockable instruction with memory operand that spans across two pages may, given some rare internal conditions, hang the system.

Implication: When this erratum occurs, the system may hang. Intel has not observed this erratum with any commercially available software or system.

Workaround: Lockable data should always be contained in a single page.

Status: For the steppings affected, see the *Summary of Table of Changes*

X3. Memory Aliasing with Inconsistent A and D Bits May Cause Processor Deadlock

Problem: In the event that software implements memory aliasing by having two page directory entries (PDEs) point to a common page table entry (PTE) and the Accessed and Dirty bits for the two PDEs are allowed to become inconsistent the processor may become deadlocked.

Implication: This erratum has not been observed with commercially available software.

Workaround: Software that needs to implement memory aliasing in this way should manage the consistency of the Accessed and Dirty bits.

Status: For the steppings affected, see the *Summary of Table of Changes*

X4. RDMSR or WRMSR to Invalid MSR Address May Not Cause GP Fault

Problem: The RDMSR and WRMSR instructions allow reading or writing of MSR's(Model Specific Registers) based on the index number placed in ECX. The processor should reject access to any reserved or unimplemented MSRs by generating #GP(0). However, there are some invalid MSR addressers for which the processor will not generate #GP(0). This erratum has not been observed with commercially available software.

Implication: For RDMSR, undefined values will be read into EDX:EAX. For WRMSR, undefined processor behavior may result.

Workaround: Do not use invalid MSR addresses with RDMSR or WRMSR.

Status: For the steppings affected, see the *Summary of Table of Changes*

X5. Unable to Disable Reads/Writes to Performance Monitoring Related MSRs

Problem: The Performance Monitoring Available bit in the miscellaneous processor features MSR (IA32_MISC_ENABLES.7) was defined so that when it is cleared to a 0, RDMSR/ WRMSR/RDPMC instructions would return all zeros for reads of and prevent any write to Performance Monitoring related MSRs. Currently it is possible to read from or write to Performance Monitoring related MSRs when the Performance Monitoring Available bit is cleared to a 0.

Implication: It is not possible to disallow reads and writes to the Performance Monitoring MSRs. Intel has not observed this erratum with commercially available software of system.

Workaround: None.

Status: For the steppings affected, see the *Summary of Table of Changes*

X6. Move to Control Register Instruction May Generate a Breakpoint Report

Problem: A move (MOV) to Control register (CR) instruction where Control register is CR0, CR3 or CR4 may generate a breakpoint report.

Implication: MOV to Control Register Instruction is not expected to generate a breakpoint report.

Workaround: Ignore breakpoint data from MOV to CR instruction.

Status: For the steppings affected, see the *Summary of Table of Changes*

X7. Error in Instruction Fetch Unit (IFU) Can Result in an Erroneous Machine Check-Exception (#MC)

Problem: A rare combination of events including the generation of a bus lock(s), the execution of a WBINVD instruction, and a page accessed or dirty bit assist may result in an erroneous Machine Check-Exception (#MC).

Implication: Due to this erratum, unexpected machine check-exception (#MC) is generated. Intel has not been able to reproduce this erratum with commercially available software.

Workaround: It is possible for BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary of Table of Changes*

X8. Code Fetch Matching Disabled Debug Register May Cause Debug Exception

Problem: The bits L0-3 and G0-3 enable breakpoints local to a task and global to all tasks, respectively. If one of these bits is set, a breakpoint is enabled, corresponding to the addresses in the debug registers DR0-DR3. If at least one of these breakpoints is enabled, any of these registers are *disabled* (i.e., L*n* and G*n* are 0), and RW*n* for the disabled register is 00 (indicating a breakpoint on instruction execution), normally an instruction fetch will not cause an instruction-breakpoint fault based on a match with the address in the disabled register(s). However, if the address in a disabled register matches the address of a code fetch which also results in a page fault, an instruction-breakpoint fault will occur.

Implication: While debugging software, extraneous instruction-breakpoint faults may be encountered if breakpoint registers are not cleared when they are disabled. Debug software which does not implement a code breakpoint handler will fail, if this occurs. If a handler is present, the fault will be serviced. Mixing data and code may exacerbate this problem by allowing disabled data breakpoint registers to break on an instruction fetch.

Workaround: The debug handler should clear breakpoint registers before they become disabled.

Status: For the steppings affected, see the *Summary of Table of Changes*

X9. Upper Four PAT Entries Not Usable With Mode B or Mode C Paging

Problem: The Page Attribute Table (PAT) contains eight entries, which must all be initialized and considered when setting up memory types for the Pentium M processor. However, in Mode B or Mode C paging, the upper four entries do not function correctly for 4-Kbyte pages. Specifically, bit 7 of page table entries that translate addresses to 4-Kbyte pages should be used as the upper bit of a 3-bit index to determine the PAT entry that specifies the memory type for the page. When Mode B (CR4.PSE = 1) and/or Mode C (CR4.PAE) are enabled, the processor forces this bit to zero when determining the memory type regardless of the value in the page table entry. The upper four entries of the PAT function correctly for 2-Mbyte and 4-Mbyte large pages (specified by bit 12 of the page directory entry for those translations).

Implication: Only the lower four PAT entries are useful for 4-KB translations when Mode B or C paging is used. In Mode A paging (4-Kbyte pages only), all eight entries may be used. All eight entries may be used for large pages in Mode B or C paging.

Workaround: None.

Status: For the steppings affected, see the *Summary of Table of Changes*

X10. SSE/SSE2 Streaming Store Resulting in a Self-Modifying Code (SMC) Event May Cause Unexpected Behavior

Problem: An SSE or SSE2 streaming store that results in a Self-Modifying Code (SMC) event may cause unexpected behavior. The SMC event occurs on a full address match of code contained in L1 cache.

Implication: Due to this erratum, any of the following events may occur:

1. A data access break point may be incorrectly reported on the instruction pointer (IP) just before the store instruction.
2. A non-cacheable store can appear twice on the external bus (the first time it will write only 8 bytes, the second time it will write the entire 16 bytes).

Intel has not observed this erratum with any commercially available software. This erratum has been seen in a synthetic test environment.

Workaround: None.

Status: For the steppings affected, see the *Summary of Table of Changes*

X11. Code Segment limit violation may occur on 4 Gigabyte limit check

Problem: Code Segment limit violation may occur on 4 Gigabyte limit check when the code stream wraps around in a way that one instruction ends at the last byte of the segment and the next instruction begins at 0x0.

Implication: This is a rare condition that may result in a system hang. Intel has not observed this erratum with any commercially available software, or system.

Workaround: Avoid code that wraps around segment limit.

Status: For the steppings affected, see the *Summary of Table of Changes*

X12. FST Instruction with Numeric and Null Segment Exceptions may cause General Protection Faults to be Missed and FP Linear Address (FLA) Mismatch

Problem: FST instruction combined with numeric and null segment exceptions may cause General Protection Faults to be missed and FP Linear Address (FLA) mismatch.

Implication: This is a rare condition that may result in a system hang. Intel has not observed this erratum with any commercially available software, or system.

Workaround: None identified.

Status: For the steppings affected, see the *Summary of Table of Changes*

X13. Code Segment (CS) is Incorrect on SMM Handler when SMBASE is not Aligned

Problem: With SMBASE being relocated to a non-aligned address, during SMM entry the CS can be improperly updated which can lead to an incorrect SMM handler.

Implication: This is a rare condition that may result in a system hang. Intel has not observed this erratum with any commercially available software, or system.

Workaround: Align SMBASE to 32K byte.

Status: For the steppings affected, see the *Summary of Table of Changes*

X14. Page with PAT (Page Attribute Table) Set to USWC (Uncacheable Speculative Write Combine) While Associated MTRR (Memory Type Range Register) is UC (Uncacheable) May Consolidate to UC

Problem: A page whose PAT memory type is USWC while the relevant MTRR memory type is UC, the consolidated memory type may be treated as UC (rather than WC as specified in IA-32 Intel® Architecture Software Developer's Manual).

Implication: When this erratum occurs, the memory page may be as UC (rather than WC). This may have a negative performance impact.

Workaround: None identified.

Status: For the steppings affected, see the *Summary of Table of Changes*

X15. Under Certain Conditions LTR (Load Task Register) Instruction May Result in System Hang

Problem: An LTR instruction may result in a system hang if all the following conditions are met:

1. Invalid data selector of the TR (Task Register) resulting with either #GP (General Protection Fault) or #NP (Segment Not Present Fault).

2. GDT (Global Descriptor Table) is not 8-bytes aligned.

3. Data BP (breakpoint) is set on cache line containing the descriptor data.

Implication: This erratum may result in system hang if all conditions have been met. This erratum has not been observed in commercial operating systems or software. For performance reasons, GDT is typically aligned to 8-bytes.

Workaround: Software should align GDT to 8-bytes.

Status: For the steppings affected, see the *Summary of Table of Changes*

X16. Loading from Memory Type USWC (Uncacheable Speculative Write Combine) May Get Its Data Internally Forwarded from a Previous Pending Store

Problem: A load from memory type USWC may get its data internally forwarded from a pending store. As a result, the expected load may never be issued to the external bus.

Implication: When this erratum occurs, a USWC load request may be satisfied without being observed on the external bus. There are no known usage models where this behavior results in any negative side-effects.

Workaround: Do not use memory type USWC for memory that has read side-effects.

Status: For the steppings affected, see the *Summary of Table of Change*

X17. FXSAVE after FNINIT Without an Intervening FP (Floating Point) Instruction May Save Uninitialized Values for FDP (x87 FPU Instruction Operand (Data) Pointer Offset) and FDS (x87 FPU Instruction Operand (Data) Pointer Selector)

Problem: An FXSAVE after FNINIT without an intervening FP instruction may save uninitialized values for FDP and FDS.

Implication: When this erratum occurs, the values for FDP/FDS in the FXSAVE structure may appear to be random values. These values will be initialized by the first FP instruction executed after the FXRSTOR that restore the saved floating point state. Any FP instruction with memory operand will initialize FDP/FDS. Intel has not observed this erratum with any commercially available software.

Workaround: After an FINIT, do not expect the FXSAVE memory image to be correct, until at least one FP instruction with a memory operand has been executed.

Status: For the steppings affected, see the *Summary of Table of Changes*

X18. FSTP (Floating Point Store) Instruction Under Certain Conditions May Result In Erroneously Setting a Valid Bit on an FP (Floating Point) Stack Register

Problem: An FSTP instruction with an PDE/PTE (Page Directory Entry/Page Table Entry) A/D bit update followed by user mode access fault due to a code fetch to a page that has supervisor only access permission may result in erroneously setting a valid bit of an FP stack register. The FP top of stack pointer is unchanged.

Implication: This erratum may cause an unexpected stack overflow.

Workaround: User mode code should not count on being able to recover from illegal accesses to memory regions protected with supervisor only access when using FP instructions.

Status: For the steppings affected, see the *Summary of Table of Changes*

§

Specification Changes

There are no specification changes for this revision of this Specification Update.

§

Specification Clarifications

There are no specification clarifications for this revision of this Specification Update.

§

Documentation Changes

There are no documentation changes for this revision of this Specification Update.

§